

Pipelinebasierte Erzeugung von Karten

Thomas Skowron

Previously

„Überblick über Rendering-
Techniken und Software“

FOSSGIS-Konferenz 2017

Status Quo

OSM Planet



PostgreSQL



Renderer

Flaschenhals



Operativer Aufwand



Flaschenhals



PostgreSQL

- + SQL
 - + ACID, MVCC & Transaktionen
 - + Indizes
 - + Rollenmanagement
 - + skriptbar
 - + failover
 - + ...
- Performance (PostGIS)
 - operativer Aufwand
 - Speicherbedarf

 Brauchen wir das Alles?

Gegenvorschlag:



Gegenvorschlag:

parse | map-reduce | render

Schritt 1

OSM Daten in Geodaten wandeln

Schritt 2
Filtern

Schritt 3

Daten mappen

Schritt 4

Ins Zielformat konvertieren

Aber wie?

Mit **Tools**, die jeweils genau
eine Sache tun

und einem
portablen Datenformat

Wie wärs mit Shapefiles?

Was bräuchte ein
geeignetes Format?

Performance

lineares Schreiben, paralleles Lesen

Skalierbar

kleine bis riesige Datensätze

Tagstruktur

Attributtabellen, nein Danke!

Zukunftskompatibel
gerüstet für Änderungen

Shapefile

Performance mäßig

Skalierbar nein, 2 GB Limit

Tagstruktur nein

Zukunftskompatibel nein

GeoJSON

Performance mäßig

Skalierbar mäßig, singlethreaded

Tagstruktur ja

Zukunftskompatibel begrenzt

GeoPackage

Performance schlecht (SQLite)

Skalierbar mäßig

Tagstruktur ja

Zukunftskompatibel ja

Performance



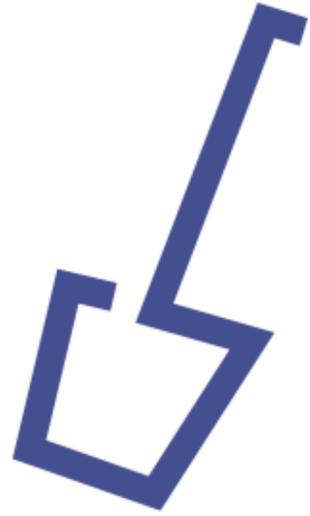
Flexibilität

Wir brauchen Mut zu Neuem.
Ohne Änderung kein Fortschritt.

Wie müsste ein neues Format aussehen?

- binär
- Blöcke, streambar
- einzelner Stream, nicht mehrere Dateien
- nicht SQLite
- nicht obskur
- offen und erweiterbar

Vorschlag



SPATEN

Basierend auf
Protocol Buffers und WKB

Offene Spezifikation unter
<https://thomas.skowron.eu/spaten/>

Referenzimplementierung in Go

github.com/thomersch/grandine/lib/spaten

Etwa 50% kleiner als GeoJSON*

* YMMV

Version 0

Feedback & Input erwünscht

Ziel?

```
grandine-spatialize -in planet.osm.pbf -mapping roads.yml |  
grandine-tiler -out tiles/roads/ -zoom 14
```

```
osmium export -f spatent planet.osm.pbf |  
  gradine-converter -mapping roads.yml |  
gradine-tiler -out tiles/roads/ -zoom 14
```

(noch nicht)

```
osmium export -f spatent planet.osm.pbf |  
  your-tool-here -fancyfy |  
  magic-renderer
```

(noch nicht)

Austauschbarkeit von Tools

Was fehlt?

Ein Markup, das
Normalisierung und
Umwandlung von Daten
vereinfacht

Beispiel

LineString(10 20, 10 21, ...)

highway=primary
max_height=11'2"



LineString(10 20, 10 21, ...)

@layer=transportation
class=primary
max_height=3.4

„Simple things should be simple, complex things should be possible.“

Alan Kay

Filterkriterium

```
- src:  
  key: railway  
  value: "*"
dest:  
- {key: "@layer", value: "transportation"}
- {key: "class", value: "railway"}
- {key: "maxspeed", value: "$maxspeed", type: int}
```

Ausgabe

Übernahme des Wertes

Typenumwandlung

Da geht noch was!

HELP NEEDED

Zukunft

Größere Flexibilität mit weniger
Programmieraufwand

Schnelleres Verarbeiten von
Daten mit weniger Hardware

Weniger „Points of Failure“

Es ist noch viel zu tun

Datenformat, Tools, Markup, ...

Lasst uns die Zukunft
gemeinsam bauen!